Discrete Applied Mathematics 156 (2008) 3267–3275

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Preemptive scheduling and antichain polyhedra

Alain Quilliot

Université BLAISE PASCAL and CNRS, LIMOS Laboratory, LIMOS, UMR CNRS 6158, Bat ISIMA, Campus des Cézeaux BP 10125, 63173 Aubière, France

ARTICLE INFO

Article history:
Received 10 October 2005
Received in revised form 22 May 2007
Accepted 22 May 2008
Available online 9 September 2008

Keywords:
Partially ordered sets
Hypergraphs
Linear programming
Multiprocessor scheduling

ABSTRACT

We present a theoretical framework, which is based upon notions of ordered hypergraphs and antichain polyhedra, and which is dedicated to the combinatorial analysis of preemptive scheduling problems submitted to parallelization constraints.

This framework allows us to characterize specific partially ordered structures which are such that induced preemptive scheduling problems may be solved through linear programming. To prove that, in the general case, optimal preemptive schedules may be searched inside some connected subset of the vertex set of an *Antichain Polyhedron*.

1. Introduction

Partially ordered (Poset) structures and hypergraphs are among the combinatorial tools which most often appear inside Discrete Mathematics and Combinatorial Optimization models.

Partially ordered sets have a large range of applications, many of them related to planning and scheduling (see [1,2,12,14,17,21,28,32]). Hypergraphs are essentially involved in the combinatorial analysis of global constraints (see [34,38]), which may appear in the modelization of information storage and retrieval problems (see [5,7,19,24,26,27]), of robotics and scheduling problems (see [1,11,25]), of resource allocation problems or even of some genetics or archeology problems (see [4,18,22]).

We first use poset and hypergraph formalisms in order to provide preemptive task scheduling problems with a general combinatorial framework. Preemptive problems have been less studied that non-preemptive ones. Still, they are important since they correspond to many practical applications (multiprocessor scheduling, grid computing, civil engineering project management, truck fleet planning). Also they may be viewed as relaxations of non-preemptive problems. Thus, capturing some of their combinatorial properties may help in designing solutions for non-preemptive problems and for mixed problems. Our combinatorial framework will allow us to obtain structural information about the complexity of these problems and about the kind of algorithms which may be designed in order to solve them.

We define here an *Ordered Hypergraph* $H = (Z, E, \rho)$ as being some finite partially ordered set (Z, ρ) given together with some subset (*edge*) family E. We call it a *Weighted Ordered Hypergraph* if every vertex $z \in Z$ is endowed with some *weight* (or length) $d(z) \in \mathbb{R}^+$. Of course, we use the Ordered Hypergraph formalism in order to represent some preemptive task sets which are required to be planned, while minimizing some cost function, and while taking into account temporal and resource constraints. The induced combinatorial optimization problems may be for instance the *Preemptive Multiprocessor Scheduling Problems* (see [10,14–16,31,32]), or the *Preemptive Resource Constrained Project Scheduling Problem* (RCPSP), (see [3,6,20,21]). Dealing with them requires the use of algorithmic tools which are suitable for the handling of partial or complete schedules inside optimization processes: local transformation procedures, bounding scheme, domination scheme .... This leads us to define, for any weighted ordered hypergraph $H = (Z, E, \rho, d)$, an *Antichain Polyhedron* Λ_H whose vertices derive from specific subsets of the edge set E. Such a polyhedron was first introduced by Sauer and Stone (see [35,36]), Papadimitriou,

E-mail address: alain.quilliot@isima.fr.

0166-218X/$ – see front matter © 2008 Published by Elsevier B.V.
doi:10.1016/j.dam.2008.05.036

Yannanakis (see [32]), in the context of non-preemptive multiprocessor scheduling problems (see [31]), and has been used (see for instance [8,9,13]) in order to get theoretical bounds for the *Resource Constraint Project Scheduling* problem.

Next, we use this framework in order to rewrite several preemptive scheduling problems involving H as search problems defined on some specific connected Subsets $\mathbf{V}\text{-}\mathbf{PR}_H$, and in order to state the most important result of this paper: this result provides us with a characterization of the case when solving a makespan minimization preemptive scheduling problem can be done in a simple way by solving a linear program defined on the polyhedron Λ_H.

2. Ordered hypergraphs: A theoretical framework for the handling of preemptive scheduling problems

Let Z be some finite (we call it *vertex set*) set, E be some (*edge set*) subset family of Z, and ρ be some binary relation defined on Z. We suppose that ρ **does not contain any circuit**. This means that ρ may be extended, through transitivity, into a partial ordering ρ_T of the set Z.

The triple $H = (Z, E, \rho)$ is then called an *Ordered Hypergraph*. We say that Z is the *vertex* set, that E is the *edge* set of H, and that ρ is the *partial ordering* of H.

Remark 1. As we shall see further, the edge set E may be defined in an implicit way, that means as a predicate function defined on the set $\boldsymbol{P}(X)$ of all the subsets of X. Of course, the algorithmic complexity of the problems which may be related to the ordered hypergraph H varies depending on the fact that H is defined in an explicit or in an implicit way.

We say that H is *monotonic*, if the two following statements are true:

- If $e \in E$ is some edge of H, then any subset of e is also in E;
- For any $x \in X$, the singleton subset $\{x\}$ is in E.

If H is monotonic, and if every vertex x of H is endowed with some positive *weight* (or length) $d(x) \in \mathbb{R}^+$, then we say that the 4-uple (X, E, ρ, d) defines a *weighted ordered hypergraph*.

We denote by ρ_T the transitive closure of ρ. A ρ-*antichain* of the ordered set (X, ρ) is a subset B of X such that: there does not exist x, y in B such that $x\rho_T y$. We say that such a subset B is a *valid antichain* of the ordered hypergraph H if it belongs to the edge set E. We denote by E_ρ the set of all valid antichains of H, and we say that E_ρ is the *Valid Antichain Set* associated with H.

We may provide the Valid Antichain Set E_ρ with an oriented graph structure by setting, for any pair e, e' in E_ρ: $e\tau e'$ iff there exists $x \in e$ and $y \in e'$ such that $x\rho y$. Then the oriented graph $K_H = (E_\rho, \tau)$ is called the Antichain Graph of H.

As we mentioned in the introduction, we will be considering any ordered hypergraph as the expression of some task system appearing in some scheduling problem, and we are going to study the way feasible "efficient" schedules may be identified with specific vertices of some polyhedron. We believe that this process will eventually lead to the emergence of linear programming based algorithms for the management of complex scheduling problems.

2.1. Basic concepts: Task representations and Antichain Polyhedron

Let $H = (X, E, \rho, d)$ be a weighted ordered hypergraph.

If $I = [\alpha, \beta]$ is some closed interval of the real line $[0, +\infty[$, we say that α is the *start-point* of I and that β is the *end-point* of I. If

$$J = \{[\alpha_1, \beta_1], [\alpha_2, \beta_2], \dots, [\alpha_n, \beta_n]\}$$

is a finite family of disjoint closed intervals of the real line $[0, +\infty[$, such that $\alpha_1 < \beta_1 < \alpha_2 < \beta_2 \cdots < \alpha_n < \beta_n$, then we say that J is a *temporal phasis*, with *start-point* equal to α_1 and with *end-point* equal to β_n.

We call *Task Representation* of $H = (X, E, \rho, d)$, any function Φ which makes correspond, to any vertex x of H, a temporal phasis $\Phi(x)$ in such a way that:

- the length (Lebesgue measure in the sense of Measure Theory) of $\Phi(x)$ is equal to $d(x)$;
- if $x, y \in X$ are such that $x\rho y$, then the start-point $\mathrm{Start}(\Phi, y)$ of $\Phi(y)$ is at least equal to the end-point $\mathrm{End}(\Phi, x)$ of $\Phi(x)$;
- for any positive real number t, the subset $e_\Phi(t)$ of X, which is defined by:

  $$e_\Phi(t) = \{x \in X \text{ such that } t \text{ belongs to the interior } \Phi(x)^\circ \text{ of } \Phi(x)\},$$

  belongs to the edge set E.

If Φ is such a task representation of H, and if e is a valid antichain of H, then we set:

$$Z(\Phi)_e = \text{Length (Lebesgue Measure) of the set } \{t \in [0, +\infty[\text{ such that } e = e_\Phi(t)\}.$$

We say that the vector $Z(\Phi) = (Z(\Phi)_e, e \in E_\rho)$ is the E_ρ indexed vector which *derives* from the task representation Φ.

We define the *makespan* of Φ as being the quantity $\mathbf{Makespan}(\Phi) = \mathrm{Max}_{x\in X}\mathrm{End}(\Phi, x)$. We say that the edge $e \in E$ is *active* for the task representation Φ if there exists some positive real number t, such that the subset

$$e_\Phi(t) = \{x \in X \text{ such that } t \text{ belongs to the interior } \Phi(x)^\circ \text{ of } \Phi(x)\}$$

is equal to e. Of course, any edge $e \in E$ which is active for Φ belongs to the valid antichain set E_ρ.

We denote by $\mathbf{TR}(H)$ the set of all task representations of H. Clearly, the fact that we suppose E to be monotonic, ensures the non-vacuity of $\mathbf{TR}(H)$.

2.1.1. *A scheduling problem related to weighted ordered hypergraphs*

The above terminology clearly tends to identify the vertex set X of the hypergraph H with some task set which should be scheduled inside the Time Space $[0, +\infty[$. According to such an interpretation, the partial ordering ρ will define an *anteriority constraint* on the task set X, the weight function d will summarize the *running times* of the tasks of X, and the edge set E will characterize the tasks which may be simultaneously run. Also, a task representation Φ is going to define a *feasible preemptive* schedule of the tasks of X. This way of linking scheduling and ordered hypergraphs leads us to set a general preemptive scheduling problem **P-SCHEDULE** as follows:

P-SCHEDULE:

{*Given:*

- *the weighted ordered hypergraph* $H = (X, E, \rho, d)$;
- *a cost real valued function* ***Cost*** *defined on the task representation set* $\mathbf{TR}(H)$;

Compute a task representation Φ, *which minimizes the cost value* $\mathbf{Cost}(\Phi)$}.

In many cases, **Cost** will be the *makespan* function. Still, one may faces variants involving deadlines and delay penalties, or specific costs C_e, $e \in E$, where every C_e is a cost per time unit induced by a simultaneous run of the tasks of the edge e. In case $\mathbf{Cost}(\Phi)$ may be written $\mathbf{Cost}(\Phi) = C.Z(\Phi)$, where $Z(\Phi)$ is the E_ρ indexed vector which derives from Φ, then we talk about the *linear version* of **P-SCHEDULE** which is induced by H and by the cost vector $C = (C_e, e \in E_\rho)$.

In many cases, E will be defined in an implicit way. For instance, if there exists some integer $k \geq 1$ such that $e \subset X$ is in E iff $\mathrm{Card}(e) \leq k$, and if $\mathbf{Cost} = \mathbf{Makespan}$, then **P-SCHEDULE** is an instance of the *Preemptive k Processor Scheduling Problem* (see [29,36]). Also, if there exists some *resource* vector V in $\mathbb{R}^n$, and some function υ which makes correspond, to any vertex x in X, some positive vector $\upsilon(x)$ of $\mathbb{R}^n$, in such a way that $e \subset X$ is in E iff $\Sigma_{x\in e}\upsilon(x) \leq V$, and if $\mathbf{Cost} = \mathbf{Makespan}$, then **P-SCHEDULE** is an instance of the *Preemptive Resource Constrained Project Scheduling Problem* (RCPSP), (see [9,23,33]).

2.1.2. *The Antichain Polyhedron* Λ_H

We call *Antichain Polyhedron* associated with the weighted ordered hypergraph $H = (X, E, \rho, d)$, the set of all feasible solutions $z = (z_e, e \in E_\rho) \geq 0$ of the following linear constraint system Λ_H:

$$\Lambda_H : \quad \left\{z = (z_e, e \in E_\rho) \geq 0, \text{ such that: for any } x \in X,\ \Sigma_{e/x\in e} z_e = d(x)\right\}.$$

As a matter of fact, we denote by Λ_H both the polyhedron and its related linear constraint system. The polyhedron Λ_H is a convex closed subset of the vector space $\mathbb{R}^{\mathrm{Card}(E_\rho)}$. In case the edge set E is defined in an implicit way, $\mathrm{Card}(E_\rho)$ may be very large. The intersection of Λ_H with the hyperspace $z_{\mathrm{Nil}} = 0$, where Nil is the empty edge, is compact, and contains all the vertices of Λ_H.

If z is some vector in Λ_H, then we say that an edge $e \in E_\rho$ is *active* for z if $z_e \neq 0$. We set $\mathrm{ACT}(z) = \{e, e \in E_\rho$, such that $z_e \neq 0\}$, and we say that $\mathrm{ACT}(z)$ is the *Active Antichain Subset* associated with z.

Let us denote by M_H the $\{0, 1\}$-matrix associated with Λ_H. The rows and the columns of M_H may respectively be identified with the vertices and with the valid antichains of H. Then Λ_H may also be rewritten: $\Lambda_H = \{z = (z_e, e \in E_\rho) \geq 0$, such that $M_H.z = d\}$.

An element z of Λ_H (see for instance [38]) is a *vertex* z of Λ_H if there exists a column subset $B \subset E_\rho$, which defines a submatrix $M_{H,B}$ of M_H such that:

- $M_{H,B}$ is invertible;
- $M_{H,B}^{-1}.d =$ the restriction z^B of z to B;
- the restriction $z^{E_\rho - B}$ of z to the edges which are not in $B \subset E_\rho$, is null.

Such a column subset $B \subset E_\rho$ is a *basis subset associated* with the vertex z. A vertex z may admits several associated basis subsets.

We denote by $V(\Lambda_H)$ the vertex set of Λ_H. It is known (see [38]), that $V(\Lambda_H)$ is endowed with a canonical adjacency relation, which comes as follows: two distinct vertices z and z' of Λ_H are adjacent if and only there exists two associated basis subsets I and I', such that the cardinality of the symmetric difference $(I \Delta I')$ is equal to 2.

2.1.3. *Linking task representations and the Antichain Polyhedron* Λ_H

Let Φ a task representation of a weighted ordered hypergraph $H = (X, E, \rho, d)$, and $Z(\Phi) = (Z(\Phi)_e, e \in E_\rho)$ be the E_ρ indexed vector which *derives* from the task representation Φ. Clearly, $Z(\Phi)$ belongs to Λ_H. In case a cost vector $C = (C_e, e \in E_\rho) \geq 0$ is given, then solving, through a column generation scheme, the linear program:

$$L_H^C : \{\textit{Compute } z \textit{ in } \Lambda_H \textit{ which minimizes the quantity } C.z\}$$

provides us with a lower bound of the linear **P-SCHEDULE** instance related to H and C (see for instance [9,23]).

Conversely, not any vector z in Λ_H may be considered as the vector $Z(\Phi)$ associated with some task representation Φ. One easily checks that this will be true if and only if the active antichain subset $\mathrm{ACT}(z) \subset E_\rho$ made with all the edges e which are active for z, does not contain any circuit in the sense of the Antichain Graph $K_H = (E_\rho, \tau)$.

This remark, combined with the fact that we expect optimal solutions of the **P-SCHEDULE** Problem to be related, in most cases, to specific vertices of Λ_H, leads us to define the following subset $\mathbf{V} - \mathbf{PR}_H$ of $V(\Lambda_H)$:

$\mathbf{V}-\mathbf{PR}_H =$ *Preemptive Vertex Subset* of the polyhedron Λ_H= {all vertices z of Λ_H which may be associated with some *basis subset* B such that the subgraph $K_H(B)$ of the Antichain Graph K_H, which is induced by B, does not contain any circuit}.

2.2. Some basic results

We are now able to state two important results.

The first one (*Linear Reformulation Theorem*), will allow us to consider any linear version of the **P-SCHEDULE** problem as a search problem defined on the *Preemptive Vertex Subset* $\mathbf{V}-\mathbf{PR}_H$. Thus, dealing with such a **P-SCHEDULE** instance will mean dealing with specific basis subsets. Though it is easy to prove, this result is non-trivial due to the fact that, even if z is some vertex of Λ_H such that ACT(z) does not admit any circuit for the graph structure K_H, then it may occur that a related basis subset B admits some circuit for the graph structure K_H. It will be used inside Section 3.

The second one (*Connectivity Theorem*) will provide us with information about the connectivity (in the sense of the adjacency relation which is defined in a natural way on the vertex set $V(\Lambda_H)$), of the subset $\mathbf{V}-\mathbf{PR}_H$, and thus will open the way to the resolution of any linear instance of the **P-SCHEDULE** problem through a well-driven walk on the set $\mathbf{V}-\mathbf{PR}_H$.

Theorem 1 (*Linear Reformulation Theorem*). *Given a weighted ordered hypergraph $H = (X, E, \rho, d)$, and a E_ρ indexed vector $C \geq 0$. Then there exists a task representation Φ_0 which is an optimal solution of the related **P-SCHEDULE** linear instance, and whose deriving vector $Z(\Phi_0)$ is in the Preemptive Vertex Subset $\mathbf{V}-\mathbf{PR}_H$.*

Proof. Let Φ be some task representation of H, let $z = Z(\Phi)$ the vector of Λ_H which derives from Φ, and let ACT(z) be the active edge subset of z. We know that the oriented subgraph $K_H(\text{ACT}(z))$ of the antichain graph K_H, which is induced by ACT(z), does not contain any circuit.

Let us set $B = \{\{x\}, x \in X\}$ = the set of all the singleton antichains. One easily checks that the subgraph $K_H(\text{ACT}(z) \cup B)$ of the antichain graph K_H, which is induced by $\text{ACT}(z) \cup B$, does not contain any circuit. Also, the rank of the restriction of the incidence matrix M_H to the $\text{ACT}(z) \cup B$ indexed columns is equal to Card(X).

It turns out that applying the Simplex Algorithm to the following linear program:

{*Find a $A(z) \cup B$ indexed vector $v \geq 0$, such that: for every $x \in X$, $\Sigma_{e\in \text{ACT}(z)\cup B/x\in e} v_e = d(x)$;*
and which minimizes the quantity $\Sigma_{e\in \text{ACT}(z)\cup B/x\in e} C_e.v_e = d(x)$},

makes appear a vertex $z°$ of the polyhedron Λ_H, which is such that:

- $C.z° \leq C.z$ = Optimal Value of **P-SCHEDULE**;
- the subgraph $K_H(A°)$ of K_H which is induced by a basis subset $A°$ associated with $z°$, does not contain any circuit. Of course, the active subset ACT($z°$) of $z°$ is included into $A°$.

Then we conclude. □

Theorem 2 (*Connectivity Theorem*). *The subset $\mathbf{V}-\mathbf{PR}_H$ is connected in the sense of the usual adjacency relation which exists between the vertices of the polyhedron Λ_H.*

Proof. Let us consider our polyhedron Λ_H, as well as some some vertex z of Λ_H which belongs to $\mathbf{V}-\mathbf{PR}_H$. The active subset associated with z is denoted by ACT(z). Let us set $B = \{\{x\}, x \in X\}$ = the set of all the singleton antichains. B is a basis subset related to a vertex z^B of Λ_H, which is defined by: for any $x \in X$, $z^B_x = d(x)$, and the rank of the restriction of the incidence matrix M_H to the $\text{ACT}(z) \cup B$ indexed columns is equal to Card(X).

So it comes that any vertex of the polyhedron $\Lambda_H(\text{ACT}(z) \cup B)$, which is defined by:

$\Lambda_H(\text{ACT}(z) \cup B)$ = {all the $\text{ACT}(z) \cup B$ indexed vectors $v \geq 0$, such that:
for every $x \in X$, $\Sigma_{e\in \text{ACT}(z)\cup/x\in e} v_e = d(x)$}

is also a vertex of Λ_H, and that if z', z'' are two adjacent vertices of $\Lambda_H(\text{ACT}(z) \cup B)$, then they are also adjacent in Λ_H. Also one easily checks that the subgraph $K_H(\text{ACT}(z) \cup B)$ of the antichain graph K_H, which is induced by $\text{ACT}(z) \cup B$, does not contain any circuit. We deduce that any vertex of $\Lambda_H(\text{ACT}(z) \cup B)$ belongs to $\mathbf{V}-\mathbf{PR}_H$.

But we know that the vertex set of any polyhedron is connected (see [37]). It turns out that there exists a path Γ between z and z^B in the vertex set of $\Lambda_H(A(z) \cup B)$, and that this path is also a path in $\mathbf{V}-\mathbf{PR}_H$. We conclude. □

What we feel is that those two results might open the way to new classes of algorithms for the management of the linear **P-SCHEDULE** problem, which may be rewritten:

{*Given:*

- *the weighted ordered hypergraph $H = (X, E, \rho, d)$;*
- *a E_ρ indexed cost vector $C = (C_e, e \in E_\rho) \geq 0$.*

Find a vertex z in $\mathbf{V}-\mathbf{PR}_H$ such that the cost $C.z$ is the smallest possible}.

P-SCHEDULE becomes a search problem defined on a connected domain, and may eventually be handled through the use of local search methods. One may notice (degeneracy of the related linear programs) that the moves related to the neighbourhood structure which is induced on $\mathbf{V}-\mathbf{PR}_H$ by the polyhedron structure of Λ_H, are most often likely not to cause any change in the value of the related *cost* value. Because of this, those elementary moves should be used as *micro-moves*, and *operational local moves* should be defined as well-driven sequences of such micro-moves.

We are not going to go further in this direction, since it is not the main purpose of this paper. Still, we may try do describe in a few words what could be the global scheme of such an algorithm. One might think for instance in driving a current vertex z of $\mathbf{V}-\mathbf{PR}_H$, together with an associated basis subset B of E_ρ and a linear extension τ^* of the subgraph $K_H(B)=(B,\tau)$, through sequences of micro-moves along the edges of the polyhedron $\mathbf{V}-\mathbf{PR}_H$. It would correspond to the following algorithmic scheme Λ-**SCHEDULE**:

Λ-**SCHEDULE** Algorithmic Scheme

Initialize $z \in \mathbf{\Lambda}-\mathbf{PR}_H$; Not Stop; Curr-Sol <- *Undefined*; Curr-Val <- *Undefined*;
While Not Stop do

Let B a basis subset of E_ρ which is associated with z;
Randomly compute a linear extension τ^* of $K_H(B)=(B,\tau)$;
Extend B into a larger edge family B^* such that the subgraph $K_H(B^*)$ of K_H which is induced by B^* admits a linear extension which is compatible with τ^*;
Solve the linear program **LP1**:

$$\{\textit{Find a } B^* \textit{ indexed vector } u \geq 0, \textit{such that, for every } x \in X,\ \Sigma_{e\in B^*/x\in e} u_e = d(x), \textit{ and which minimizes the quantity } C.u\}.$$

Let z' be an optimal solution of **LP1**, and let $B' \subset B^*$ be an associated basis subset;
If $C.z' <$ Current-Value then

Curr-Sol $\leftarrow z'$; Curr-Val $\leftarrow C.z'$;

Test (z', B', τ^*) and decide whether or not to replace z by z'.

3. Commutative ordered hypergraphs: A case when optimal preemptive scheduling can be done through linear programming

We are now going to focus on the case of preemptive scheduling, and on the way it is possible to use the linear program:

L^1_H : {Compute z in Λ_H which minimizes the quantity $\mathbf{1}.z$}

in order to solve the following preemptive scheduling problem:

PM-SCHEDULE : {*Find a task representation Φ of H, whose makespan is minimal*}.

As a matter of fact, this section will be devoted to a result which identifies ordered hypergraphs H such that, for any weight function d, the respective optimal values of L^1_H and **PM-SCHEDULE** are equal. This result, which extends a former result by Moukrim and Quilliot (see [29,30]) related to the *k Processor Scheduling Problem*, is going to involve a specific property of ordered hypergraphs, which we will call *Commutativity Property*.

3.1. Commutative ordered hypergraphs

For any pair (e,e') in the antichain edge set E_ρ, we set:

- $\mathrm{MAX}(e,e') = \{x \in e\cup e'$ such that there exists y in $e\cup e'$ which satisfies $x\rho_T y\}$;
- $\mathrm{MIN}(e,e') = \{y \in e\cup e'$ such that there exists x in $e\cup e'$ which satisfies $x\rho_T y\}$;
- $\mathrm{EQ}(e,e') = (e\cup e') - (\mathrm{MAX}(e,e')\cup \mathrm{MIN}(e,e'))$.

We say that the weighted ordered hypergraph $H=(X,E,\rho,d)$ is *weakly commutative* if, for any pair (e,e') in the antichain edge set E_ρ, it is possible to find $f_1..f_q$, and $f'_1..f'_p$ in E_ρ, such that:

(C1) for any $x\in \mathrm{EQ}(e,e')$,

$$\frac{1}{q}\mathrm{Card}(\{i=1..q \text{ such that } x\in f_i\}) + \frac{1}{p}\mathrm{Card}(\{j=1..p \text{ such that } x\in f'_j\}) = 1;$$

(C2) for any $i=1..q$, $f_i \subset \mathrm{EQ}(e,e')$; for any $j=1..p$, $f'_j \subset \mathrm{EQ}(e,e')$;
(C3) for any $i=1..q$, $\mathrm{MIN}(e,e')\cup f_i \in E_\rho$; for any $j=1..p$, $\mathrm{MAX}(e,e')\cup f'_j \in E_\rho$.

In such a case, we denote by COMMUTE(e, e') the subset of E_ρ which is defined by:

$$\text{COMMUTE}(e, e') = \left\{\text{MIN}(e, e') \cup f_i, i = 1..q\right\} \cup \left\{\text{MAX}(e, e') \cup f'_j, j = 1..p\right\},$$

and we notice that COMMUTE(e, e') does not contain any circuit in the sense of the antichain graph $K_H = (E_\rho, \tau)$.

If, for any pair (e, e') as above, $f_1..f_q$, and $f'_1..f'_p$ may be chosen in such a way that one of the two following assertions (C4) or (C5) is true:

(C4) there exists j in $\{1..p\}$ such that $f'_j \subset \text{EQ}(e, e') \cap e'$;
(C5) there exists i in $\{1..q\}$ such that $f_i \subset \text{EQ}(e, e') \cap e$;

then we say that H is *commutative*. In case (C4) is true, then we say that e is the *receiver of the commutation of* e *and* e', else we say that e' is the *receiver of this commutation*.

If it is possible, for any pair (e, e') as above, to do in such a way that $p = q = 1$, then we say that H is *strongly commutative*.

If $e, e' \in E_\rho$ are such that both above families $F = \{f_1..f_k\}$, and $F' = \{f'_1..f'_p\}$ exist, then we say that e and e' are *(weakly, strongly) commuting*, and that the antichains MIN$(e, e') \cup f_i$, $i = 1..k$, and MAX$(e, e') \cup f'_j$, $j = 1..p$, result from *(weakly, strongly) commuting* e and e' through F and F'.

Clearly, the *strong commutativity* implies the *commutativity*, which in turn implies the *weak commutativity*.

In case there exists a number k such that E is the set of all the subsets of X with no more than k elements (case of the *k Processor Scheduling Problem*), then we see that H is commutative if for every pair (e, e') in E_ρ, neither MAX(e, e') nor MIN(e, e') has more that k elements. In such a case, H is also strongly commutative. One also checks that, in any case, if the partial ordering ρ defines an interval order (see [30]), then the ordered hypergraph $H = (X, E, \rho)$ is strongly commutative.

We can now state our main result, which extends a previous result (see [29,30]) dedicated to the case of the *k Processor Scheduling Problem*, and which tells us that if the weighted ordered hypergraph $H = (X, E, \rho, d)$ is commutative, then the optimal value of the related **PM-SCHEDULE** may be computed by solving the linear program L^1_H.

Theorem 3 (*Exactness Theorem*). *Let* $H = (X, E, \rho, d)$ *be some weighted ordered hypergraph. Then the following statements* (D1) *and* (D2) *are true:*

(D1) *If, for any integral weight function* $d \geq 0$*, there exists a task representation of* H *whose makespan is equal to the optimal value of* L^1_H*, then* H *is weakly commutative;*
(D2) *If* H *is commutative, then, for any integral weight function* $d \geq 0$*, there exists a task representation of* H *whose makespan is equal to the optimal value of* L^1_H.

Proof. Preliminary: If $\{e_1, \ldots, e_p\}$ is a subset of E_ρ such that $p = \text{Card}(X)$, then $\{e_1, \ldots, e_p\}$ defines a square submatrix of the constraint matrix M_H of L^1_H, and the determinant of this submatrix is called a subdeterminant of M_H. Let us denote by δ the largest common multiplier of all the subdeterminants of M_H, and let set $\varepsilon = 1/\delta$. We call ε the *index value* of M_H. The integral weight function d may be written $d = \varepsilon.D$, where D is also an integral function. Linear Programming Theory tells us (see [37]) that any vertex z of Λ_H may be written $z = z^\circ.\varepsilon$, where z° is an integral vector. It comes that there exists an integer Q such that: Optimal Value of $L^1_H = Q.\varepsilon$, and, because of the *Linear Reformulation Theorem*, that an optimal solution Φ of **PM-SCHEDULE** may be chosen in such a way that the vector $z = Z(\Phi)$ which derives from Φ is also the product of ε by an integral vector.

Proof of (D1): Let us consider e and e' in E_ρ, together with some weight function d such that:

- $d(x) = 1$ for every $x \in e \cup e'$;
- $d(x) = 0$, for any $x \in X$ which is not in $e \cup e'$.

This provides us with a situation such that:

- the optimal value of the linear program L^1_H is equal to 2;
- any task representation Φ of H must be such that: for any $x \in \text{MIN}(e, e')$, $y \in \text{MAX}(e, e')$, x precedes y according to Φ, which means that $\text{End}(\Phi, x) \leq \text{Start}(\Phi, y)$.

Let us consider a task representation Φ of H such that:

- the makespan of Φ is equal to 2 (that means to the optimal value of L^1_H);
- the deriving vector $z = Z(\Phi) = z^\circ.\varepsilon$ is a vertex of Λ_H.

Then we may define two subsets F and F' of E_ρ by setting:

- f is in F, if there exists u in E_ρ, such that u is active for Φ and such that u may be written $u = f \cup \text{MIN}(e, e')$. We count f, inside F, as many times as the integral quantity z°_u;
- f' is in F', if there exists u' in E_ρ, such that u' is active for Φ and such that u' may be written $u' = f' \cup \text{MAX}(e, e')$. We count f', inside F', as many times as the integral quantity $z^\circ_{u'}$.

We see that, according to this construction, the task representation Φ may be viewed as resulting from the weak commutation of e and e' through F and F'. We deduce our result.

Proof of (D2): Let us suppose that H is commutative and that d is a given integral weight function. For any subset U of X, let us set:

$$\mathrm{INF}(U) = \{x \in U \text{ such that there does not exists } y \text{ in } U \text{ which satisfies } y\rho_T x, \\ \text{where } \rho_T \text{ denotes the transitive closure of } \rho\}.$$

Then we notice that:

(E1) For any pair $a, b \in E_\rho$ such that $b \subset \mathrm{INF}(X - a)$, we have:
(E1′) For any $x \in b$, the subset $\{y \in X$ such that $y\rho_T x\}$ of X is included into $a \cap \mathrm{INF}(X)$;
(E1″) $b \subset \mathrm{INF}(X) \cup \mathrm{INF}(X - \mathrm{INF}(X))$.

We may, for any subset U of X, define $\mathrm{SUP}(U)$ by the same way, and proceed to the same remark.

Let us consider now some vertex $z = z^\circ.\varepsilon$ which is an optimal solution of the linear program L^1_H, with value $\mathbf{1}.z = Q.\varepsilon$.

We define a *co-subset* W of E_ρ as being a subset inside which any element may eventually appear several times. If W is such a co-subset, and if $e \in E_\rho$, then we denote by $O(W, e)$ the number of times e appears inside W, and we call this number the *occurrence number* of e in W. The cardinality of such a co-subset becomes equal to the sum of all the occurrence numbers $O(W, e)$. The union of two co-subsets W' and W'' is a co-subset W such that, for any e in E_ρ, $O(W, e) = O(W', e) + O(W'', e)$.

Let $W(z)$ be the active antichain subset of z: $W(z) = \{e \in E_\rho$ such that $z_e \neq 0\}$.

As a matter of fact, we may consider $W(z)$ as a co-subset of E_ρ, in such a way that for any valid antichain $e \in W(z)$, the occurrence number of e in $W(z)$ is equal to the integral number z°_e. We know that $W(z)$ may contain some circuit in the sense of the antichain graph $K_H = (E_\rho, \tau)$, and that such a situation keeps us from converting in a straightforward way z into a task representation of H. Still $W(z)$ provides us with the existence of a co-subset $W = W(z)$ such that:

(E2) $\mathrm{Card}(W) = Q$;
(E3) For any $x \in X$, the number of antichains of W which contain x is equal to $D(x) = \delta.d(x)$.

What we are now going to prove is that if W is a co-subset such that (E2) and (E3) are true, then there exists a co-subset W° of E_ρ such that (E2) and (E3) are true and such that W° does not contain any circuit in the sense of the antichain graph $K_H = (E_\rho, \tau)$. In case $W = W(z)$, we will conclude by deducing from W° the existence of a task representation of H with makespan equal to $Q.\varepsilon$. So let us consider some co-subset W which satisfies (E2) and (E3), and let us proceed by induction on Q.

Let us consider some valid antichain e in W. If, for any $x \in e$, we set $d^\circ(x) = d(x) - \varepsilon$, and if, for any $y \notin e$, we set $d(y) = d^\circ(y)$, then we turn H into a weighted ordered hypergraph $H^\circ = \mathrm{REDUCE}(H, e)$ in such a way that:

- the optimal value of L^1_{H*} is equal to $\varepsilon.(Q - 1)$;
- the weight function d^* remains equal to the product of ε with an integral function D^*;
- H^* remains commutative.

The induction hypothesis may be applied to H^*, and enables us to conclude in the cases when $e \subset \mathrm{INF}(X)$ or $e \subset \mathrm{SUP}(X)$. Thus, we only need to prove the existence of a co-subset W_1 which satisfies (E2) and (E3), and which contains some antichain e such that either $e \subset \mathrm{INF}(X)$ or $e \subset \mathrm{SUP}(X)$.

Let us suppose now that some antichain $e \in W$ is such that $e \cap \mathrm{INF}(X) = \emptyset$. Then we apply the induction to the hypergraph $H^\circ = \mathrm{REDUCE}(H, e)$, and we get some co-subset W'_1 such that: (E4)

- W'_1 contains some antichain e' which is such that: $e' \subset \mathrm{INF}(X - e)$;
- $\mathrm{Card}(W'_1) = Q - 1$;
- For any $x \in X$, the number of valid antichains e in W'_1 which contain x is equal to $D(x) - 1 = \delta.(d(x) - \varepsilon)$ if $x \in e$ and is equal to $D(x) = \delta.d(x)$ else.

But, in such a case, we deduce from $e \cap \mathrm{INF}(X) = \emptyset$ that $e' \subset \mathrm{INF}(X)$ and the result. It comes that, for any antichain e in W, we may suppose that $e \cap \mathrm{INF}(X)$ is not empty, as well as (symetry) $e \cap \mathrm{SUP}(X)$.

In the same way, we see that we may suppose that there exists $u \in W$ such that $u \subset \mathrm{INF}(X) \cup \mathrm{INF}(X - \mathrm{INF}(X))$. If it is not true, then we pick up $e \in W$, we apply the induction hypothesis to $H^* = \mathrm{REDUCE}(H, e)$, and we get W'_1 such that (E4) above is true. We deduce that there exists u in W'_1 such that $u \subset \mathrm{INF}(X - e)$, which also means such that $u \subset \mathrm{INF}(X) \cup \mathrm{INF}(X - \mathrm{INF}(X))$. Then we only need to replace W by $\{e\} \cup W'_1$ in order to get the result.

By considering W and u as above, and by applying the induction hypothesis to $H^* = \mathrm{REDUCE}(H, u)$, we get W'_1 such that (E4) above is true. By replacing W by $\{u\} \cup W'_1$, we see that we may do in such way that W and $u \in W$ satisfy: (E5)

* $u \subset \mathrm{INF}(X) \cup \mathrm{INF}(X - \mathrm{INF}(X))$.
* W may be written $W = \{u\} \cup W'_1$ in such a way that:
  - W_1 does not contain any circuit in the sense of the antichain graph $K_H = (E_\rho, \tau)$;
  - $\mathrm{Card}(W'_1) = Q - 1$;
  - For any $x \in X$, the number of antichains u in W'_1 which contain x is equal to $D(x) - 1 = \delta.(d(x) - \varepsilon)$ if $x \in u$ and is equal to $D(x) = \delta.d(x)$ else.
* $u \cap \mathrm{INF}(X - \mathrm{INF}(X))$ is minimal for the inclusion relation.

Since W_1' does not contain any circuit in the sense of the relation τ, we may label the elements of W_1' by setting $W_1' = v_1..v_{Q-1}$ in such a way that: for any $i, j = 1..Q-1$, $v_i \tau v_j$ implies $i < j$.

In case there does not exist i in $1..Q-1$ such that $v_i \tau u$, then we are done, else we may pick up i_o in $1..Q-1$, which is such that $v_{io} \tau u$, and which is the largest possible with this property. If $i_o < Q-1$, then we may conclude by applying the induction hypothesis on X, E and ρ and on the weight function d^* which is defined, for any $x \in X$, by:

$$d^*(x) = d(x) - \mathrm{Card}(N(x)).\varepsilon,$$

where $N(x)$ is the set $N(x) = \{i = i_\circ + 1, \ldots, Q-1, \text{ such that } x \in v_i\}$.

So we suppose now (non-trivial case), that $i_\circ = Q-1$, and we try to commute u and $v = v_{Q-1}$.

This means that we make appear $f_1..f_k$, and $g_1..g_p$ in E_ρ, such that:

- for any $x \in \mathrm{EQ}(u, v)$,
$$\frac{1}{k}\mathrm{Card}(\{i = 1..k \text{ such that } x \in f_i\}) + \frac{1}{p}\mathrm{Card}(\{j = 1..p \text{ such that } x \in g_j\}) = 1;$$
- for any $i = 1..k, f_i \subset \mathrm{EQ}(u, v)$ and for any $j = 1..p, g_j \subset \mathrm{EQ}(u, v)$;
- for any $i = 1..k, j = 1..p$, $\mathrm{MIN}(u, v) \cup f_i$ belongs to E_ρ and $\mathrm{MAX}(u, v) \cup g_j$ belongs to E_ρ;
- one of the two following assertions (E6.1) or (E6.2) is true:

(E6.1) there exists j in $\{1..p\}$ such that $g_j \subset \mathrm{EQ}(u, v) \cap v$;
(E6.2) there exists i in $\{1..k\}$ such that $f_i \subset \mathrm{EQ}(u, v) \cap u$.

Two cases must be considered:

First case: (E6.1) above is true (u is the *receiver of the commutation of* u *and* v).

We may suppose that $g_1 \subset \mathrm{EQ}(u, v) \cap v$, and we set $h_1 = f_1 \cup \mathrm{MAX}(u, v)$. In such a case, we deduce from (E5) that any element in h_1 is also in $\mathrm{SUP}(X)$, since any element of $\mathrm{MAX}(u, v)$ is in $\mathrm{SUP}(X)$, as well as any element of $\mathrm{EQ}(u, v) \cap v$. Also, we may deduce some optimal solution z^* of L_H^1 by setting, for any valid antichain e in E_ρ:

$$A_e = \mathrm{Card}(\{s = 1..Q-2 \text{ such that } e = v_s\});$$
$$B_e = \mathrm{Card}(\{i = 1..k \text{ such that } e = f_i \cup \mathrm{MAX}(u, v)\});$$
$$C_e = \mathrm{Card}(\{j = 1..p \text{ such that } e = g_j \cup \mathrm{MAX}(u, v)\});$$
$$z_e^* = \varepsilon.\left[A_e + \frac{1}{k}B_e + \frac{1}{p}C_e\right].$$

This solution is such that $z_{h1}^* \neq 0$. The vector z^* may not be a vertex of the polyhedron Λ_H, but we know that it can be expressed as a barycentric combination of vectors which are all vertices of Λ_H and which are at the same time optimal solutions of the linear program L_H^1. That means that there exists a vertex z' of Λ_H, which is an optimal solution of L_H^1, and which is such that $z_{h1}^* \neq 0$. It becomes sufficient to replace z by z' in order to conclude.

Second case: (E6.2) above is true (v is the *receiver of the commutation of* u *and* v).

We may suppose that $f_1 \subset \mathrm{EQ}(u, v) \cap u$ and we set $h_1 = f_1 \cup \mathrm{MIN}(u, v)$. In such a case, we deduce from (E5) that $h_1 \subset \mathrm{INF}(X - \mathrm{INF}(X))$ and that $h_1 \cap \mathrm{INF}(X - \mathrm{INF}(X))$ is strictly included into $u \cap \mathrm{INF}(X - \mathrm{INF}(X))$. (E7)

By proceeding by the same way as for the first case, we see that there must exist a vertex z' of Λ_H, which is an optimal solution of L_H^1, and which is such that $z_{h1}^* \neq 0$. We conclude by deducing from (E7) a contradiction on the minimality of $u \cap \mathrm{INF}(X - \mathrm{INF}(X))$. □

Remark 2. The above proof is non-constructive. Still, one may easily check that, in case the ordered hypergraph H is strongly commutative, then the last arguments (E6.1) and (E6.2), which involve barycentric combinations of vertices of Λ_H, may be removed. In such a case, the proof of the *Exactness Theorem* gives rise to a recursive reconstruction algorithm which takes in input an optimal solution of the linear program L_H^1 and which turns it into an optimal solution of the **PM-SCHEDULE** problem. This reconstruction algorithm may be adaptated to the general case, which means to the case when H is not commutative. Then we get a heuristic reconstruction procedure, which turns any feasible solution of L_H^1 into a feasible solution of **PM-SCHEDULE**, and which preserves optimality in case H is strongly commutative.

Remark 3. One may ask if the (D2) assertion of the *Exactness Theorem* is true in case H is only weakly commutative, which means in case H only satisfies properties (C1), (C2), (C3). As a matter of fact, we are not able to answer this question. Our proof of the *Exactness Theorem* requires any commutation of two antichains to involve a receiver. But one may propose as a conjecture that the *Exactness Theorem* may be turned into an equivalence only involving weak commutativity.

Remark 4. The above proof is non-constructive, and one may ask if it is possible to replace the linear programming based arguments (E6.1) and (E6.2) by combinatorial arguments only involving antichain surgery. It is probably possible, but the related proof might not be simpler. Our linear programming argument helps us in dealing with the fact that commuting two antichains in the co-subset W is likely both to increase the size of W and to yield a new solution of L_H^1 which may not be related to vertices of the polyhedron Λ_H. It allows us to be sure that, at any time during our proof, the current co-subset W is such that there exists a solution z of L_H^1, whose active subset is W, and which satisfies, for any antichain e in W: $z_e = \varepsilon.z_e^\circ$, where z_e° is equal to the occurrence number of e in W.

4. Conclusion

We have introduced a combinatorial framework involving hypergraphs and partially ordered sets, in order to make it easier to deal with scheduling preemptive problems. We used it in order to get theoretical results related to the structure of the related feasible sets, and in order to state a result which identifies cases when linear programming is enough to deal with those scheduling problems.

Still, we neither really dealt with algorithms nor with the complexity. So we shall end by setting several questions. What about the complexity of the preemptive scheduling problems which derive from a commutative ordered hypergraph? How can we design an efficient algorithm which would turn an optimal solution of L_H^1 into an optimal solution of **PM-SCHEDULE** in case the ordered hypergraph H is commutative? Also, is weak commutativity enough in order to get the part (D2) of the *Exactness Theorem*?

References

[1] J.F. Allen, Towards a general theory of action and time, Artificial Intelligence 23 (1984) 123–154.
[2] K.R. Baker, Introduction to Sequencing and Scheduling, Wiley, N.Y, 1974.
[3] P. Baptiste, Resource constraints for preemptive and non-preemptive scheduling, MSC Thesis, University PARIS VI, 1995.
[4] S. Benzer, On the topology of the genetic fine structure, Proc. Acad. Sci. USA 45 (1959) 1607–1620.
[5] C. Berge, Graphes et Hypergraphes, Dunod, 1975.
[6] J. Blazewiecz, K.H. Ecker, G. Schmidt, J. Weglarz, Scheduling in Computer and Manufacturing Systems, 2nd ed., Springer-Verlag, Berlin, 1993.
[7] K.S. Booth, J.S. Lueker, Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms, J. Comput. Sci. 13 (1976) 335–379.
[8] P. Brucker, S. Knust, A linear programming and constraint propagation based lower bound for the RCPSP, European J. Oper. Res. 127 (2000) 355–362.
[9] P. Brucker, S. Knust, A. Schoo, O. Thiele, A branch and bound algorithm for the resource constrained project scheduling problem, European J. Oper. Res. 107 (2) (1998) 272–288.
[10] J. Carlier, P. Chretienne, Problèmes d'ordonnancements: Modélisation, complexité et algorithmes, Masson, Paris, 1988.
[11] M. Carter, A survey on practical applications of examination timetabling algorithms, Oper. Res. 34 (2) (1986) 193–202.
[12] M. Chein, M. Habib, The jump number of Dags and posets: An introduction, Ann. Discrete Math. 9 (1980) 189–194.
[13] E. Demeulemeester, W. Herroelen, New benchmark results for the multiple resource constrained project scheduling problem, Manage. Sci. 43 (1997) 1485–1492.
[14] D. Dolev, M.K. Warmuth, Scheduling precedence graphs of bounded heights, J. Algorithms 5 (1984) 48–59.
[15] D. Dolev, M.K. Warmuth, Profile scheduling of opposing forests and level orders, SIAM J. Algorithms Discrete Methods 6 (1985) 665–687.
[16] P. Duchet, Problèmes de représentations et noyaux, Thèse d'Etat PARIS VI, 1981.
[17] B. Dushnik, W. Miller, Partially ordered sets, Amer. J. Math. 63 (1941) 600–610.
[18] D.R. Fulkerson, J.R. Gross, Incidence matrices and interval graphs, Pacific J. Math. 15 (1965) 835–855.
[19] S.P. Ghosh, File organization: The consecutive retrieval property, Comm. ACM 9 (1975) 802–808.
[20] R.L. Grahamson, E.L. Lawler, J.K. Lenstra, A.H.G Rinnooy-Kan, Optimization and approximation in deterministic sequencing...: A survey, Ann. Discrete Math. 5 (1979) 287–326.
[21] W. Herroelen, E. Demeulemeester, Bert de Reyck, A classification scheme for project scheduling, in: Project Scheduling: Recent Models, Algorithms and Applications, Kluwer Acad Publishers, 1999, pp. 1–26.
[22] D.G. Kindall, Incidence matrices, interval graphs and seriation in archeology, Pacific J. Math. 28 (1969) 565–570.
[23] R. Kolisch, A. Sprecher, A. Drexel, Characterization and generation of a general class of resource constrained project scheduling problems, Manage. Sci. 41 (10) (1995) 1693–1703.
[24] L.T. Kou, Polynomial complete consecutive information retrieval problems, SIAM J. Comput. 6 (1992) 67–75.
[25] E.L. Lawler, K.J. Lenstra, A.H.G. Rinnoy-Kan, D.B. Schmoys, Sequencing and scheduling: Algorithms and complexity, in: S.C. Graves, A.H.G. Rinnoy-Kan, P.H. Zipkin (Eds.), Handbook of Operation Research and Management Sciences, Vol. 4: Logistics of Production and Inventory, North-Holland, Amsterdam, 1993, pp. 445–522.
[26] F. Luccio, F.P. Preparata, Storage for consecutive retrieval, Inform. Process. Lett. 5 (3) (1976) 68–71.
[27] A. Mingozzi, V. Maniezzo, S. Ricciardelli, L. Bianco, An exact algorithm for project scheduling with resource constraints based on a new mathematical formulation, Manage. Sci. 44 (1998) 714–729.
[28] R.H. Mohring, F.J. Rademacher, Scheduling problems with resource duration interactions, Methods Oper. Res. 48 (1984) 423–452.
[29] A. Moukrim, A. Quilliot, A relation between multiprocessor scheduling and linear programming, Order 14 (1998) 269–278.
[30] A. Moukrim, A. Quilliot, Optimal preemptive scheduling on a fixed number of identical parallel machines, Oper. Res. Lett. 33 (2) (2005) 143–150.
[31] R.R. Muntz, E.G. Coffman, Preemptive scheduling of real time tasks on multiprocessor systems, J. ACM 17 (2) (1970) 324–338.
[32] C.H. Papadimitriou, M. Yannanakis, Scheduling interval ordered tasks, SIAM J. Comput. 8 (1979) 405–409.
[33] J.H. Patterson, A comparison of exact approaches for solving the multiple constrained resource project scheduling problem, Manage. Sci. 30 (7) (1984) 854–867.
[34] A. Quilliot, Sun Xiao, Algorithmic characterization of interval ordered hypergraphs and applications, Discrete Appl. Math. 51 (1994) 159–173.
[35] N. Sauer, M.G. Stone, Rational preemptive scheduling, Order 4 (1987) 195–206.
[36] N. Sauer, M.G. Stone, Preemptive scheduling of interval orders is polynomial, Order 5 (1989) 345–348.
[37] A. Schrijver, Theory of Linear and Integer Programming, Wiley Inter, NY, 1986.
[38] P. Van Hentenryk, Constraint Programming, North Holland, 1997.